Diverse Routing with the star property

Jean-Claude Bermond, David Coudert, Gianlorenzo D'Angelo,
Fatima Zahra Moataz

RESEARCH
REPORT
N° 8071
September 2012
Project-Team MASCOTTE

ISSN 0249-6399 ISRN INRIA/RR--8071--FR+ENG

Diverse Routing with the star property

Jean-Claude Bermond[*], David Coudert[†], Gianlorenzo D'Angelo[‡], Fatima Zahra Moataz[§]

Project-Team MASCOTTE

Research Report n° 8071 — September 2012 — 23 pages

Abstract: The notion of Shared Risk Link Groups (SRLG) has been introduced to capture survivability issues where a set of links of a network fail simultaneously. In this context, the *diverse routing* problem is to find a set of SRLG-disjoint paths between a given pair of end nodes of the network. This problem has been proved NP-complete in general [25] and some polynomial instances have been characterized [12].
In this paper, we investigate the diverse routing problem in networks satisfying the *star property* [29]. This property states that an edge may be subject to several SRLGs but all edges subject to a given SRLG are incident to a common node. We first provide counter-examples to the polynomial time algorithm proposed in [29] for computing pairs of SRLG-disjoint paths in networks with the star property, and then prove that this problem is in fact NP-hard in the strong sense. More generally, we prove that the diverse routing problem in networks with the star property is NP-hard in the strong sense, hard to approximate, and $W[1]$-hard when the parameter is the number of SRLG-disjoint paths. Last, we devise polynomial time algorithms for practically relevant subcases, in particular when the number of SRLG is constant, the maximum degree of the vertices is strictly smaller than 5, and when the network is a directed acyclic graph.

Key-words: Diverse routing, Shared Risk Link Group, colored graph, labeled graph, complexity, algorithms.

[*] MASCOTTE Project, INRIA/I3S(CNRS/UNSA), France. Jean-Claude.Bermond@inria.fr
[†] MASCOTTE Project, INRIA/I3S(CNRS/UNSA), France. david.coudert@inria.fr
[‡] MASCOTTE Project, INRIA/I3S(CNRS/UNSA), France. gianlorenzo.d_angelo@inria.fr
[§] MASCOTTE Project, INRIA/I3S(CNRS/UNSA), France.

RESEARCH CENTRE
SOPHIA ANTIPOLIS – MÉDITERRANÉE

2004 route des Lucioles - BP 93
06902 Sophia Antipolis Cedex

Diverse Routing with the star property

Résumé : La notion de groupe de liens partageant un risque (*Shared Risk Link Group*, (SRLG) a été introduite pour modéliser des problèmes de tolérance aux pannes simultanées d'ensembles de liens d'un réseau. Dans ce contexte, le problème du *routage diversifié* est de trouver un ensemble de chemins SRLG-disjoints entre une paire donnée de nœuds du réseau. Ce problème a été prouvé NP-complet en général [25] et certains cas polynomiaux ont été caractérisés [12].

Dans cet article, nous étudions le problème du routage diversifié dans les réseaux satisfaisants la *propriété d'étoile* [29]. Dans un réseau satisfaisant la propriété d'étoile, un lien peut être affecté par plusieurs SRLGs, mais tous les liens affectés par un même SRLG sont incidents à un même sommet. Nous commençons par donner des contre-exemples à l'algorithme polynomial proposé dans [29] pour le calcul de paires de chemins SRLG-disjoints dans les réseaux satisfaisants la propriété d'étoile. Puis, nous prouvons que ce problème est en fait NP-difficile au sens fort. Plus généralement, nous montrons que le problème du routage diversifié dans les réseaux avec la propriété d'étoile est NP-difficile au sens fort, APX-difficile, et W [1]-difficile lorsque le paramètre est le nombre de chemins SRLG-disjoints. Enfin, nous caractérisons de nouvelles instances polynomiales, en particulier lorsque le degré maximum des sommets est 4, ou lorsque le réseau est acyclique.

Mots-clés : Diverse routing, Shared Risk Link Group, colored graph, labeled graph, complexity, algorithms.

1 Introduction

The notion of Shared Risk Link Groups (SRLG) has been introduced to capture survivability issues when a set of links of a network may fail simultaneously. More precisely, a SRLG is a set of network links that are likely to fail simultaneously if a given event (i.e., a risk) occurs. The scope of this concept is very broad, from the set of fiber links of an optical backbone network that are physically buried at the same location and so that could be cut simultaneously (i.e., *backhoe fade*), or that are located in the same seismic area, to radio links in access and backhaul networks subject to localized environmental conditions affecting signal transmission, or to traffic jam propagation in road networks.

The graph theoretic framework for studying optimization problems in networks with SRLGs is the *colored-graph* model [15, 34, 18, 16, 29, 12], also called *labeled-graph* model [11, 28, 33, 5, 7, 9, 8, 22, 21, 26, 6, 14]. In the colored-graph model, the network topology is modeled by a graph $G = (V, E)$ and the set of SRLGs by a set of colors $\mathcal{R}$. Each SRLG is modeled by a distinct color, and that color is assigned to all the edges corresponding to the network links subject to this SRLG. Also, an edge modeling a network link subject to several SRLGs will be assigned as many colors as SRLGs. A colored-graph is therefore defined by the triple $(V, E, \mathcal{R})$ in which the set $\mathcal{R}$ of colors is a covering of the edges of E.

In the context of colored graphs, basic graph connectivity problems have been re-stated in terms of colors and proved much more difficult to address than their usual counterparts. For instance, the minimum color st-path problem is to find a path from vertex s to vertex t in the graph that minimizes the cardinality of the union of the colors of the edges along that path. This problem has been proved NP-hard [31, 5] and hard to approximate [12, 21] in general, $W[2]$-hard when parameterized by the number of used colors and $W[1]$-hard when parameterized by the number of edges of the path [19]. The minimum color spanning tree problem, which asks to find a spanning tree of the graph that minimizes the cardinality of the union of the colors of its edges, has been proved NP-hard [11, 33, 26, 21], as hard to approximate as *set cover* [28], and $W[2]$-hard when parameterized by the number of used colors [19]. Other graph connectivity problems have been studied in colored-graphs such as the minimum color cut [18, 12], the minimum color st-cut [12], the minimum color maximum matching [19], and the diverse routing problem [25]. In this paper, we are interested in the last problem.

The *diverse routing* problem is one of the most fundamental problem for network survivability. It asks to find a set of color-disjoint paths between a given pair of end nodes of the graph. This problem has been proved NP-complete in general [25] even when seeking for only two color-disjoint paths [13]. But, when both the set $\mathcal{R}$ of colors is a partition of the edges and the subgraphs induced by each color are stars then the diverse routing problem can be solved in polynomial time [15]. More generally, the diverse routing problem can be solved in polynomial time when $\mathcal{R}$ is a partition and the subgraphs induced by each color are connected [12]. However, no other polynomial instances are know so far.

An interesting property when studying connectivity problems in colored graphs is the *star property* [29]. We say that a colored graph has the star property if the induced subgraph of each color is a star. From a networking perspective, this property indicates that correlated failures are sets of links incident to a router node. The failure of the router node itself is modeled with a color assigned to all its incident links. As said above, the diverse routing problem can be solved in polynomial time in colored graphs satisfying the star property and such that the set of colors forms a partition of the edges [15, 12]. Moreover, it has been proved in [13] that the minimum color st-path problem can be solved in polynomial time in colored graphs with the star property, without any assumption on the set of colors ($\mathcal{R}$ is a covering). A natural question is thus if the diverse routing problem can be solved in polynomial time in colored graphs with the star property when leveraging the partition assumption. Indeed, this question has been addressed in [29] for computing two color-disjoint paths, claiming that this can be done in polynomial time. However, in this paper we disprove the correctness of the algorithm proposed in [29], and we show that the problem is in fact NP-hard in the strong sense in this setting.

In detail, given a colored graph $G = (V, E, \mathcal{R})$, we denote by α the *maximum number of colors per arc*, by β the *maximum number of arcs having the same color*, and by Δ and Δ_c, the *maximum degree* and the *maximum colored degree*, respectively.

We show that finding k SRLG disjoint paths is NP-hard in the strong sense, even with the star property and α and β are fixed with $\alpha \geq 3$ and $\beta \geq 3$ or $\alpha \geq 6$ and $\beta \geq 2$. Moreover, unless $P = NP$, such problem cannot be approximate within $O(|V|)$ even if β is fixed, $\beta \geq 2$ and it is APX-hard if α is fixed. Such inapproximability results hold also when the graph is a Directed Acyclic Graph (DAG) or it is an undirected graph.

The above inapproximability results imply that, if α is unbounded, we cannot find an approximation factor better than $\frac{|V|}{k}$, where k is a constant. This corresponds to find k color-disjoint st-paths, where k is a given constant. In other words, the only way to cope with this problem (in a general graph with the star property) is to find a *fixed* number of color-disjoint paths. However, we can also prove that the problem is $W[1]$-hard which means that no FPT algorithms can be devised, unless $FPT = W[1]$, that is cannot exist an algorithm which finds k color-disjoint paths in $O(2^k \cdot poly(|V|))$ time. Finally we have shown that even finding a fixed number $k \geq 2$ of color-disjoint paths is NP-hard. This implies that it is impossible to devise an algorithm which finds k color-disjoint paths in $O(|V|^{O(poly(k))})$ time. This last result holds even if Δ is fixed with $\Delta \geq 6 + k$; α, β and Δ_c are fixed with $\alpha \geq 4$, $\beta \geq 2$, and $\Delta_c \geq 14 + k$ or $\alpha \geq 2$, $\beta \geq 4$ and $\Delta_c \geq 2 + k$.

On the positive side, we show that the problem can be solved in polynomial time in particular subcases which are relevant in practice. Namely, we solve the problem when the input graph is a DAG, when the number of SRLGs is bounded by a constant (i.e. $|\mathcal{R}| = O(1)$) or when the degree of the graph is strictly smaller than 5. In detail, the algorithm for DAGs requires $O((|V|\Delta)^{2k})$ time. As finding k color-disjoint paths is $W[1]$-hard, this is basically the best time possible up to a constant in the exponent. The algorithm for fixed $|\mathcal{R}|$ and the one for $\Delta < 5$ require $O(|V|+|E|)$ time. Also in this case the computational time matches the theoretical lower bound. Note also that the above restrictions on the input instances do not constitute a constraint in practical applications as in real networks both the number of SRLGs and the maximum degree is usually very small.

2 Notation and problem statement and preliminaries

We model a network as a directed graph $G = (V, E)$, where the vertices in V represent the nodes and the arcs in E represent the links. We associate a color to each SRLG. Let us denote by $\mathcal{R}$ the set of all the colors, by $E(c)$ the set of arcs having color $c \in \mathcal{R}$, and by $\mathcal{R}(e)$ the set of colors associated with arc $e \in E$.

A *multi-colored graph* is a triple $mG = (V, E, \mathcal{R})$, where (V, E) is a directed graph and $\mathcal{R}$ is a set of colors assigned to E. We denote by $\alpha = \max_{e \in E} |\mathcal{R}(e)|$ (i.e. the *maximum number of colors per arc*) and by $\beta = \max_{c \in \mathcal{R}} |E(c)|$ (e.i. the *maximum number of arcs having the same color*). Given a vertex v, $\Gamma^-(v)$ and $\Gamma^+(v)$ denote the in-neighbors and out-neighbors of v, respectively, $\Gamma(v) = \Gamma^-(v) \cup \Gamma^+(v)$. A color is *incident* to a vertex v if such color is assigned to an arc incident to v. The *colored degree* of a given vertex v is its number of incident colors and it is denoted by $d_c(v)$. The *degree* of a vertex v is the size of $\Gamma(v)$ and it is denoted by $d(v)$. The *maximum degree* and the *maximum colored degree* of a graph are denoted by Δ and Δ_c, respectively.

We assume that the graphs have the so-called *star property* [29]. A color $c \in \mathcal{R}$ is called a *star color* if all arcs of $E(c)$ are incident to the same vertex. A multi-colored graph has the star property if it has only star colors.

Given a multi-colored graph mG and two vertices s and t, an st-path is an alternating sequence of vertices and arcs, beginning with s and ending with t, in which each arc is incident with the vertex immediately preceding it and the vertex immediately following it. A path is denoted by the sequence of vertices or by the sequence of vertices and arcs. We say that two paths P_1 and P_2 are *color-disjoint* if $(\cup_{e \in P_1} \mathcal{R}(e)) \cap (\cup_{e \in P_2} \mathcal{R}(e)) = \emptyset$.

We aim at finding the maximum number of color-disjoint paths, formally:

Definition 1 (Max Diverse Colored st-Paths (`MDCP`)) *Given a multi-colored graph mG and two vertices s and t, find the maximum number of color-disjoint st-paths.*

The corresponding decision problem is the following.

Definition 2 (Diverse Colored st-Paths (`DCP`)) *Given a multi-colored graph mG, two vertices s and t and an integer k, are there k color-disjoint st-paths?*

We refer the problem in which k is fixed and not in the input of the problem as k-`DCP`.

Definition 3 (k Diverse Colored st-Paths (k-`DCP`)) *Given a multi-colored graph mG and two vertices s and t, are there k color-disjoint st-paths?*

Note that, as is `MDCP` an optimization version of `DCP`, an algorithm for a problem implies an algorithm for the other one.

3 Counterexample to Luo&Wang's algorithm

In [29] it is proposed a polynomial time algorithm to find two color-disjoint paths from a source vertex s to a destination vertex t in an undirected multi-colored graph with the star property. In this section, we show that the algorithm given in [29] is not correct by giving two counterexamples that show that, on the one hand, the algorithm sometimes is not able to find two disjoint paths when they exist and on the other hand, it might find paths which are not color-disjoints.

We mainly use the notation of the paper. We denote by s the source vertex (called v_s or v_0 in the paper) and t the destination vertex (called v_d in the paper). The Luo&Wang's algorithm is a modified version of the Bhandari's algorithm [3]. In detail, the algorithm works in five steps. In the first step, a shortest path $P_a = \{s \equiv u_0, u_1, \ldots, u_\ell \equiv t\}$ from s to t is computed using e.g. Dijkstra's algorithm. In the second step, each edge $\{u_i, u_{i+1}\}$, $i = 0, 1, \ldots, \ell - 1$, in P_a is transformed into a directed arc (u_{i+1}, u_i), i.e. the arc is directed towards the opposite direction of path P_a. In the third step, some virtual vertices are added to divide edges which are not part of P_a but have their endpoints in P_a. Finally, a second path P_b from s to t in this modified graph is computed. If P_b and P_a do not share any edge or vertex, then two color-disjoint paths are found. Otherwise, the algorithm performs the segment deletion and exchange process of the Bhandari's algorithm.

In the counterexamples we mention only some colors. Other colors are not represented, we can either suppose there are no other colors or that they are different from the colors explicitly mentioned and appear on only one edge. All edges have cost 1.

In the example 1 (see Figure 1), color c is shared between edges $\{s, w_0\}$ and $\{s, v_1\}$ and color $c' \neq c$ is shared between edges $\{s, w_0\}$ and $\{s, v_2\}$. The shortest path is $P_a = \{s, w_0, t\}$. Let us apply the algorithm. As it is also explained in page 451 lines 9–10, the initialization part guarantees that the first edge of P_b, is color-disjoint from P_a. Therefore, in the example 1, the

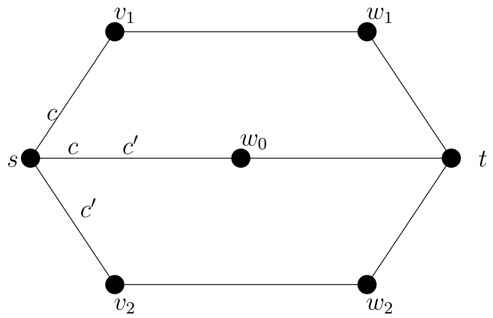


Figure 1: Example 1.

algorithm does not find any edge to start and so terminates and the authors conclude there are no two color-disjoint paths. However such two color-disjoint paths clearly exist, namely they are $P_1 = \{s, v_1, w_1, t\}$ and $P_2 = \{s, v_2, w_2, t\}$. Similar counterexample exist with the colors at destination vertex putting color d shared between edges $\{w_0, t\}$ and $\{w_1, t\}$ and color d' shared between edges $\{w_0, t\}$ and $\{w_2, t\}$.

In the example 2 (see Figure 2), we have 4 specific colors c and $c' \neq c$ (forming a star in v) and d and d' (the last two are different from c and c', but can be the same). The shortest path from s to t is $P_a = \{s, a, v, b, t\}$. In this example vertex v is a cut-vertex, then if there exist two color-disjoint paths, they both have to contain v. In this case, one path should use the subpath u, v, b and the other one should use the subpath u', v, b' (b and b' can be interchanged). Indeed, if we use the edge $\{a, v\}$ of P_a there is no path which is color-disjoint from this edge. In vertex w one path uses the edge $\{w, u\}$ and the other $\{w, u'\}$. If $d = d'$, then do not exist in example 2 two color-disjoint paths. If $d \neq d'$ then we have two color-disjoint paths $P_1 = \{s, a, z, w, u, v, b, t\}$ and $P_2 = \{s, a', w, u', v, b', t\}$.

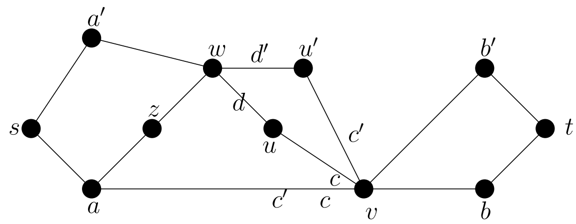


Figure 2: Example 2.

Let us apply the algorithm of [29]. The shortest path found in the first step is $P_a = \{s, a, v, b, t\}$. Procedure in Fig. 6 of [29] will find successively a', w, u, u' by case 1 and put them in V_n. It will add (case 2) v, keeping the conflict and so next step it will have to go backwards (case 3) till a with T_2 information. At that stage, if we follow strictly what is said in the algorithm we can find perhaps z but never w again and the algorithm stops missing the fact there exist two color-disjoint paths when $d \neq d'$. Another interpretation of the algorithm (or a variant of the figure by adding an edge $\{a', v\}$ with both colors c and c') will be to use the virtual edges around v and it might be that the algorithm discovers the two paths P_1 and P_2. But in all cases in w the algorithm considers only possible conflicts with P_a and there are no such conflicts. So in that situation, the algorithm gives that P_1 and P_2 are color-disjoint, but they are not if $d = d'$.

4 Hardness results

We first show that `DCP` is NP-hard in the strong sense and that `MDCP` is hard to approximate within some bound, even with some restrictions on α and β. Note that all the following results are based on transformations which use undirected graphs which can be easily reformulated for directed graphs.

In the next theorem, we give a reduction from the exact cover by 3-sets (`X3C`) problem.

Definition 4 (Exact cover by 3-sets (`X3C`)) *Given a set X with $|X| = 3q$ and a collection C of 3-element subsets of X, does C contain an exact cover for X, i.e., a sub-collection $C' \subseteq C$ such that every element of X occurs in exactly one member of C'?*

Problem `X3C` is known to be NP-hard in the strong sense [20] and it remains NP-hard if no element occurs in more than three subsets. Moreover, note that any feasible sub-collection C' has cardinality q, i.e, $|C'| = q$.

Theorem 1 `DCP` *is NP-hard in the strong sense, even with the star property and α and β are fixed with $\alpha \geq 3$ and $\beta \geq 3$ or $\alpha \geq 6$ and $\beta \geq 2$.*

Proof.
Given and instance of `X3C`, we define an instance of `DCP` as follows.

- for each element C_i of C, we define a vertex v_i, moreover, we add two vertices s and t;
- we add edges $\{s, v_i\}$ and $\{v_i, t\}$, for each $C_i \in C$;
- for each $C_i, C_j \in C$, such that $i \neq j$ and $C_i \cap C_j \neq \emptyset$, we add a new color to $\{s, v_i\}$ and $\{s, v_j\}$; every other edge has a different color (or no colors if it is allowed);
- $k = q$.

The transformation is polynomial time computable and the star property holds. See Figure 3 for two examples. By definition each color is associated with two edges and hence $\beta = 2$. As `X3C` is NP-hard even if no element occurs in more than three subsets, each edge has at most 6 colors and hence $\alpha = 6$. Furthermore, we can modify the transformation as follows: instead of adding colors as in the third item above, we define a color r_x for each element $x \in X$ and, for each $C_i = \{x, y, z\} \in C$, we associate colors r_x, r_y, and r_z to edge $\{s, v_i\}$. In this way, each edge has 3 colors and each color is associated with at most 3 edges, that is $\alpha = 3$ and $\beta = 3$.

We first show that if the answer to the instance of `X3C` is "yes", then the answer to the instance of `DCP` is "yes". Let us denote as $C' = \{C_{i_1}, C_{i_2}, \ldots, C_{i_q}\}$ an exact cover of the instance of `X3C`. We define $k = q$ paths $P_j = \{s, v_{i_j}, t\}$, for each $j = 1, 2, \ldots q$. The set of paths is denoted by P. As $|P| = q = k$, it remains to show that the paths in P are color-disjoint. By contradiction, let us assume that two paths $P_a, P_b \in P$ are not color-disjoint. As we chose different colors for each edge $\{v_i, t\}$, P_a and P_b have to share some colors in edges $\{s, v_{i_a}\}$ and $\{s, v_{i_b}\}$ which implies that $C_{i_a} \cap C_{i_b} \neq \emptyset$, a contradiction to the fact that C' is an exact cover.

We now show that if the answer to the instance of `DCP` is "yes", then the answer to the instance of `X3C` is "yes". Let us denote as P a set of k color-disjoint paths st-paths in the instance of `DCP`. Each path P_j in P is in the form $\{s, v_{i_j}, t\}$, for $j = 1, 2, \ldots, k$. We define C' as $C' = \{C_{i_1}, C_{i_2}, \ldots, C_{i_k}\}$. All the elements of C' are pairwise disjoint, in fact if by contradiction, we assume that there exist C_{i_a} and C_{i_b} such that $C_{i_a} \cap C_{i_b} \neq \emptyset$, then the corresponding paths P_a and P_b would share some colors which is a contradiction to the fact that the paths in P are color-disjoint. Moreover, C' is a covering of X as $k = q$, $|C_{i_j}| = 3$ and the sets in C' are pairwise disjoint, which implies that C' contains $3q = |X|$ distinct elements. $\square$

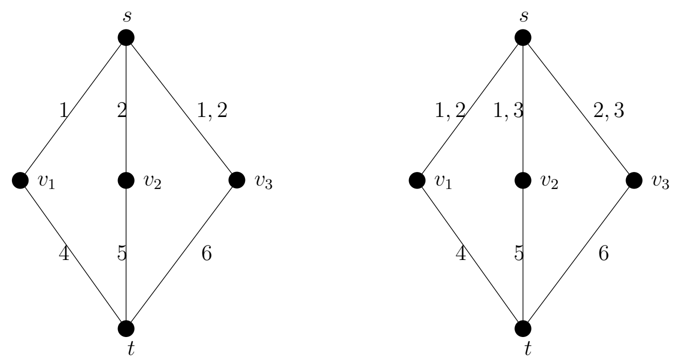


Figure 3: Examples where $C = \{C_1 = \{a,b,c\}, C_2 = \{d,e,f\}, C_3 = \{a,b,d\}\}$ (left) and $C = \{C_1 = \{a,b,d\}, C_2 = \{b,c,d\}, C_3 = \{b,e,f\}\}$ (right). Colors are represented as numbers close to edges.

In the next theorem we give an approximation factor preserving reduction from Maximum Set Packing (MSP).

Definition 5 (Maximum Set Packing (MSP)) *Given a set X and a collection C of subsets of X, find the maximum cardinality set packing, i.e., a collection of disjoint sets $C' \subseteq C$ such that $|C'|$ is maximized.*

Problem MSP is equivalent to maximum clique under PTAS-reduction, where $|C|$ corresponds to the number of vertices in the graph [1], hence it is not approximable within $O(|C|)$. Moreover, if the cardinality of all sets in C is upper bounded by a constant $c \geq 3$ or the number of occurrences in C of any element upper-bounded by a constant $B \geq 2$, then the problem is APX-complete [2, 27],

Theorem 2 MDCP *is as hard to approximate as* MSP*, with $|V| = |C|$, even if the star property holds.*

Proof. Given an instance $I_{\texttt{MSP}}$ of MSP we show that there exits a polynomial time algorithm f which transforms $I_{\texttt{MSP}}$ into an instance $I_{\texttt{MDCP}}$ of MDCP and a polynomial time algorithm g which transforms a solution $S_{\texttt{MDCP}}$ for $I_{\texttt{MDCP}}$ into a solution $S_{\texttt{MSP}}$ for MSP such that

$$OPT(I_{\texttt{MSP}}) \leq OPT(I_{\texttt{MDCP}}) \tag{1}$$

$$OBJ(S_{\texttt{MDCP}}) \leq OBJ(S_{\texttt{MSP}}), \tag{2}$$

where OPT and OBJ denote the value of the optimal solution and the value of the objective function of a solution, respectively. Inequalities 1 and 2 imply that $\frac{OPT(I_{\texttt{MSP}})}{OBJ(S_{\texttt{MSP}})} \leq \frac{OPT(I_{\texttt{MDCP}})}{OBJ(S_{\texttt{MDCP}})}$. Therefore, any α approximation algorithm for MDCP implies an α approximation algorithm for MSP.

We use the same transformation given in Theorem 1 as algorithm f.

To show inequality 1, let us consider an optimal solution C'_{OPT} for MSP, $C'_{OPT} = \{c_{i_1}, c_{i_2}, \ldots, c_{i_{|C'_{OPT}|}}\}$. We define a solution for $I_{\texttt{MDCP}}$ as the set P of paths $P_j = \{s, v_{i_j}, t\}$, for each $j = 1, 2, \ldots, |C'_{OPT}|$. It can be shown that the paths in P are color-disjoint by using the same arguments as in the proof of Theorem 1. It follows that P is a feasible solution for $I_{\texttt{MDCP}}$ with $|P| = |C'_{OPT}|$, and then, $OPT(I_{\texttt{MSP}}) = |C'_{OPT}| = |P| \leq OPT(I_{\texttt{MDCP}})$.

To show inequality 2, let us define algorithm g as the one which transforms a set P of paths in the form $\{s, v_{i_j}, t\}$, $j = 1, 2, \ldots, |P|$ (which is a solution for $I_{\texttt{MDCP}}$) into a solution C' for $I_{\texttt{MSP}}$, where $C' = \{c_{i_1}, c_{i_2}, \ldots, c_{i_{|P|}}\}$. Again, it can be show that the elements of C' are pairwise disjoint by using the same arguments used in the proof of Theorem 1 and then C' is a feasible solution for MSP such that $|P| = |C'|$. It follows that $OBJ(S_{\texttt{MDCP}}) = |P| = |C'| \leq OBJ(S_{\texttt{MSP}})$. □

Corollary 1 *Unless $P = NP$,* MDCP *cannot be approximate within $O(|V|)$ even if β is fixed, $\beta \geq 2$. Moreover, it is APX-hard if α is fixed, $\alpha \geq 3$.*

Proof. The first statement follows from the $O(|C|)$ inapproximability of MSP [1] and from the transformation where each color is associated with two edges which implies that $\beta = 2$. Moreover, as MSP is APX-hard if the cardinality of all sets in C is upper bounded by a constant $c \geq 3$ [27], we can modify the transformation as we did in Theorem 1 that is we define a color r_x for each element $x \in X$ and, for each $c_i = \{x, y, z\} \in C$, we associate colors r_x, r_y, and r_z to edge $\{s, v_i\}$. In this way, $\alpha = 3$. □

Theorem 3 MDCP *is $W[1]$-hard where the parameter is the number of color-disjoint paths, even with the star property.*

Proof. It is enough to observe that the reductions used in Theorems 1 and 2 are parameterized-preserving reductions where the parameter is the number of color-disjoint paths which corresponds to the number of disjoint subsets in a set packing. In [17, 30] it has been shown that MSP is $W[1]$-hard if the parameter is the number of disjoint subsets. □

Corollary 2 MDCP *is not in FPT, unless $FPT = W[1]$.*

As $|V|$ is an upper bound to any optimal solution of MDCP, Corollary 1 implies that, if α is unbounded, we cannot find an approximation factor better than $\frac{|V|}{c}$, where c is a constant. This corresponds to find c color-disjoint st-paths, where c is a given constant. In other words, the only way to cope with this problem (in a general graph with the star property) is to find a *fixed* number of color-disjoint paths. However, Theorem 3 implies that no FPT algorithms can be devised, unless $FPT = W[1]$, that is cannot exist an algorithm which finds k color-disjoint paths in $O(2^k \cdot poly(|V|))$ time. Moreover, the next theorem shows that even finding a fixed number $k \geq 2$ of color-disjoint paths is NP-hard. This implies that it is impossible to devise an algorithm which finds k color-disjoint paths in $O(|V|^{O(poly(k))})$ time. Note also that the transformation used in Theorems 1–3 makes use of simple graph that can be easily turned on directed acyclic graphs.

The proof of the next theorem consists in a reduction from the problem of finding a T-compatible path (or a path avoiding forbidden transitions), proved NP-complete in [32].

Let $G = (V, E)$ be an undirected graph. A transition in v is a pair of edges incident to v. A set $T(v)$ of admissible (or allowed) transitions in v is associated with vertex v in V. The transitions are represented in [32] by a graph also denoted $T(v)$ where vertices represent the edges of G incident to v and two vertices are joined if the edges they represent form an admissible transition. We call transition system the set $T = \{T(v) \mid v \in V\}$. Let $G = (V, E)$ be a graph and T a transition system, a path $P = \{v_0, e_1, v_1, \ldots, e_k, v_k\}$ in G, where $v_i \in V$, $e_i \in E$, is said to be T-compatible if, for every $1 \leq i \leq k-1$, the pair of edges $\{e_i, e_{i+1}\}$ is admissible, i.e. $\{e_i, e_{i+1}\} \in E(T(v_i))$.

Definition 6 (T-Compatible path (T-CP)) *Given a graph $G = (V, E)$, two vertices s, t in V, and a transition system T, does G contain a T-compatible path?*

It has been proved in [32] that problem T-`CP` is NP-complete in the strong sense and it remains NP-complete for simple graphs where vertices have either degree 4 or 3, vertices s and t have degree 3, and where for each vertex v:

- If $d(v) = 4$, $T(v)$ consists of two pairs of distinct edges $\{e, f\}$ and $\{g, h\}$ where e, f, g and h are the 4 edges incident to v (i.e. $T(v) = K_2 + K_2$, see Figure 4a);
- If $d(v) = 3$, $T(v)$ consists of two pairs of edges $\{e, h\}$ and $\{f, h\}$ where e, f and h are the 3 edges incident to v (i.e. $T(v) = P_3$, see Figure 4b).

(a) $K_2 + K_2$ (b) P_3

Figure 4: Transition graphs used in the transformation.

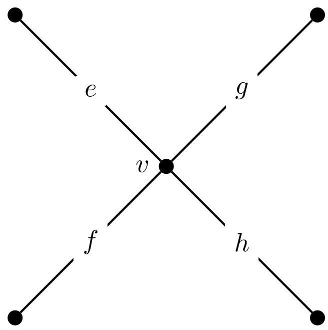
Theorem 4 *k-`DCP` is NP-hard in the strong sense for any fixed constant $k \geq 12$, even if:*

- *the star property holds;*
- *the maximum degree Δ is fixed with $\Delta \geq 6 + k$;*
- *α, β and Δ_c are fixed with $\alpha \geq 4$, $\beta \geq 2$, and $\Delta_c \geq 14 + k$ or $\alpha \geq 2$, $\beta \geq 4$ and $\Delta_c \geq 2 + k$.*

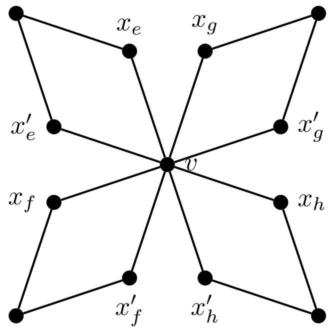
Proof. We first prove the statement for $k = 2$ and then we extend it for any fixed $k \geq 3$.

Given and instance of T-`CP`, we define an instance of 2-`DCP` as follows. We define a graph $H = (V_H, E_H, \mathcal{R})$ where:

- For every edge $e = \{u, v\}$ in G, we associate in H two paths of length 2: $\{u, x_e, v\}$ and $\{u, x'_e, v\}$. H has then $|V(G)| + 2|E(G)|$ vertices and $4|E(G)|$ edges.
- We assign the colors to edges incident to a vertex v in H as follows:
  - no colors are assigned to edges incident to t;
  - for each pair of edges e and f incident to s in G such that $f \neq e$, we assign colors C_{ef} and $C_{ef'}$ ($C_{e'f}$ and $C_{e'f'}$, resp.) to edge $\{s, x_e\}$ ($\{s, x'_e\}$, resp.), and colors C_{ef} and $C_{e'f}$ ($C_{ef'}$ and $C_{e'f'}$, resp.) to edge $\{s, x_f\}$ ($\{s, x'_f\}$, resp.);
  - for each $v \neq s, t$, and for each pair of edges e and f incident to v in G such that $f \neq e$ and $\{e, f\}$ is not an admissible transition (i.e. $\{e, f\} \notin E(T(v))$), we assign colors C_{ef} and $C_{ef'}$ ($C_{e'f}$ and $C_{e'f'}$, resp.) to edge $\{v, x_e\}$ ($\{v, x'_e\}$, resp.), and colors C_{ef} and $C_{e'f}$ ($C_{ef'}$ and $C_{e'f'}$, resp.) to edge $\{v, x_f\}$ ($\{v, x'_f\}$, resp.). As each vertex has either degree 3 or 4, two cases can occur:
    * If $d(v) = 4$, let e, f, g and h be the 4 edges incident to v and $E(T(v)) = \{\{e, f\}, \{g, h\}\}$. Then edge $\{v, x_e\}$ receives the 4 colors $C_{eg}, C_{eg'}, C_{eh}, C_{eh'}$. Similarly, edge $\{v, x_f\}$ receives the 4 colors $C_{fg}, C_{fg'}, C_{eh}$ and $C_{fh'}$. See Figure 5 for the complete list of colors assigned to all the edges.



(a) v in G, $T(v) = \{e, f\} \cup \{g, h\}$



(b) v in H

Edge	Colors
$\{v, x_e\}$	$C_{eg}, C_{eg'}, C_{eh}, C_{eh'}$
$\{v, x'_e\}$	$C_{e'g}, C_{e'g'}, C_{e'h}, C_{e'h'}$
$\{v, x_f\}$	$C_{fg}, C_{fg'}, C_{fh}, C_{fh'}$
$\{v, x'_f\}$	$C_{f'g}, C_{f'g'}, C_{f'h}, C_{f'h'}$
$\{v, x_g\}$	$C_{eg}, C_{e'g}, C_{fg}, C_{f'g}$
$\{v, x'_g\}$	$C_{eg'}, C_{e'g'}, C_{fg'}, C_{f'g'}$
$\{v, x_h\}$	$C_{eh}, C_{e'h}, C_{fh}, C_{f'h}$
$\{v, x'_h\}$	$C_{eh'}, C_{e'h'}, C_{fh'}, C_{f'h'}$

Figure 5: Color assignment for vertices with degree 4.

* If $d(v) = 3$, let e, f and h be the 3 edges incident to v and let $E(T(v)) = \{\{e, h\}, \{f, h\}\}$, then edge $\{v, x_e\}$ receives the 2 colors C_{ef} and $C_{ef'}$, edge $\{v, x_h\}$ receives no colors and so on. See Figure 6 for the complete list of colors assigned to all the edges.

The transformation is polynomial time computable and the star property holds. Moreover, note that each edge has at most 4 colors, each color is associated with two edges, the degree of each vertex is at most 8 and the color degree is at most 16. It follows that $\alpha = 4$, $\beta = 2$, $\Delta = 8$, and $\Delta_g = 16$.

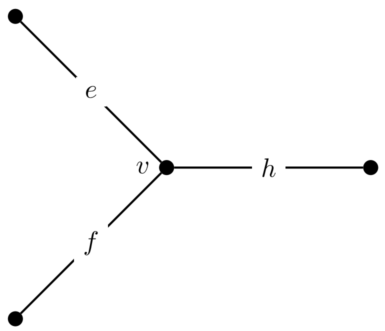
To prove the theorem, we first give the following properties.

Property 1 *Given an edge e incident to s in G, the edge $\{s, x_e\}$ in H shares a color with all the other edges incident to s but $\{s, x'_e\}$. Said otherwise, the only pair of edges incident to s having no color in common are of the form $\{\{s, x_e\}, \{s, x'_e\}\}$ for some e.*

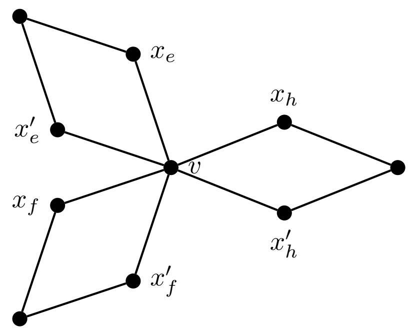
Property 2 *If $v \neq s, t$, $d(v) = 3$ and $E(T(v)) = \{\{e, h\}, \{f, h\}\}$, then only the edges $\{v, x_e\}$ and $\{v, x'_e\}$ share colors with edges $\{v, x_f\}$ and $\{v, x'_f\}$.*

Property 3 *If $v \neq s, t$, $d(v) = 4$ and $E(T(v)) = \{\{e, f\}, \{g, h\}\}$, then the edges $\{v, x_e\}$ and $\{v, x'_e\}$ ($\{v, x_g\}$ and $\{v, x'_g\}$, resp.) share no colors and share colors with any other edge but $\{v, x_f\}$ and $\{v, x'_f\}$ ($\{v, x_h\}$ and $\{v, x'_h\}$, resp.).*

We first show that if there exists a T-compatible path in G, then there exist two color-disjoint paths in H. Let $P = \{s, e_1, v_1, \ldots, e_k, v_k, e_{k+1}, t\}$ be a T-compatible path from s to t in G. $Q = \{s, x_{e_1}, v_1, \ldots, x_{e_k}, v_k, x_{e_{k+1}}, t\}$ and $Q' = \{s, x'_{e_1}, v_1, \ldots, x'_{e_k}, v_k, x'_{e_{k+1}}, t\}$ are then



(a) v in G, $T(v) = \{e, h\} \cup \{f, h\}$



(b) v in H

Edge	Colors
$\{v, x_e\}$	$C_{ef}, C_{ef'}$
$\{v, x'_e\}$	$C_{e'f}, C_{e'f'}$
$\{v, x_f\}$	$C_{ef}, C_{e'f}$
$\{v, x'_f\}$	$C_{ef'}, C_{e'f'}$
$\{v, x_h\}$	no colors
$\{v, x'_h\}$	no colors

Figure 6: Color assignment for vertices with degree 3.

two color-disjoint paths in H. In fact, any edge $\{v, x_e\}$ has no color in common with $\{v, x'_e\}$ by Properties 1, 2, or 3.

Conversely, we now show that if there exist two color-disjoint paths in H, then there exists a T-compatible path in G. Let $Q = \{s, x_1, v_1, \ldots, x_k, v_k, x_{k+1}, t\}$ and $Q' = \{s, y_1, u_1, \ldots, y_h, u_h, y_{h+1}, t\}$ be two color-disjoint paths in H. We prove by induction on $i \in \{1, \ldots, k+1\}$, that $\{x_i, y_i\} = \{x_{e_i}, x'_{e_i}\}$, $v_i = u_i$ and $k = h$.

For $i = 1$, by property 1, $\{s, x_1\}$ and $\{s, y_1\}$ have no color in common only if $\{x_1, y_1\} = \{x_e, x'_e\}$ and then $v_1 = u_1$.

Let us suppose that it is true till $i = l$ and prove it for $i = l + 1$. In $v_l = u_l$, let the two edges used by Q and Q' be $\{x_{e_l}, v_l\}$ and $\{x'_{e_l}, v_l\}$.

- If $d(v_l) = 4$, by property 3, the only possibility as Q and Q' are color-disjoint is that they use the edges $\{v_l, x_{e_{l+1}}\}$ and $\{v_l, x'_{e_{l+1}}\}$ where $\{e_l, e_{l+1}\} \in E(T(v_l))$ and so the statement is true for $i = l + 1$.

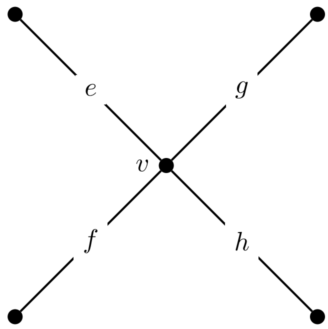
- If $d(v_l) = 3$, we distinguish two cases:
  - e_l belongs to an edge of $T(v_l)$ say $\{e_l, h_l\}$ and the paths Q and Q' being color-disjoint can only use the edges $\{v_l, x_{h_l}\}$ and $\{v_l, x'_{h_l}\}$;
  - e_l belongs to two edges in $T(v_l)$, $\{e_l, f_l\}$ and $\{e_l, h_l\}$. If one path uses the edge $\{v_l, x_{h_l}\}$ ($\{v_l, x'_{h_l}\}$, resp.) the other path cannot use the edge $\{v_l, x_{f_l}\}$ or $\{v_l, x'_{f_l}\}$ by Property 2, it has then to use edge $\{v_l, x'_{h_l}\}$ ($\{v_l, x_{h_l}\}$, resp,).

It follows that the path $P = \{s, e_1, v_1, \ldots, e_k, v_k, e_{k+1}, t\}$ satisfies $\{e_i, e_{i+1}\} \in E(T(v_i))$ for every $i \in \{1, k\}$ and then it is T-compatible.

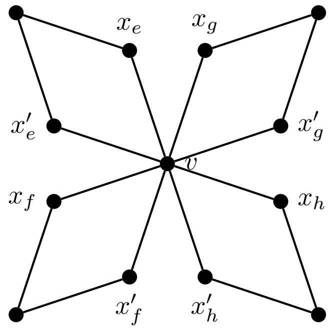
To show that the problem remains NP-hard even for fixed $\alpha \geq 2$, $\beta \geq 4$ and $\Delta_c \geq 4$, let us modify the above transformation by using the following color assignments.

- Edges incident to vertices whose degree in G is 4 have the color assignment of Figure 7;
- Edges incident to vertex s have the color assignment of Figure 8;
- the other vertices of degree 3 in G and t keep the same color assignment as before.

The above proof works with this color assignment. It follows then that 2-DCP is NP-hard even for fixed $\alpha \geq 2$, $\beta \geq 4$ and $\Delta_c \geq 4$.



(a) v in G, $T(v) = \{e, f\} \cup \{g, h\}$



(b) v in H

Edge	Colors
$\{v, x_e\}$	C_1, C_2
$\{v, x'_e\}$	C_3, C_4
$\{v, x_f\}$	C_1, C_2
$\{v, x'_f\}$	C_3, C_4
$\{v, x_g\}$	C_1, C_3
$\{v, x'_g\}$	C_2, C_4
$\{v, x_h\}$	C_1, C_3
$\{v, x'_h\}$	C_2, C_4

Figure 7: Color assignment for a vertex of degree 4 ($\neq t$).

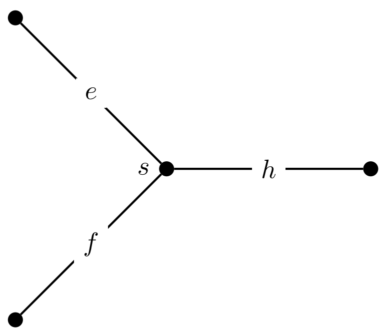
To extend the proof to the case of $k \geq 3$, it is enough to add $k-2$ paths length 2 from s to t called $P_i = \{s, w_i, t\}$, $i = 3, 4, \dots k$, with a new color assigned to each edge $\{s, w_i\}$. These paths are color-disjoint among them and with the two paths in the above transformation. Moreover, this assignment does not change α and β and increases Δ and Δ_c by $k-2$. □

As finding k color-disjoint paths is NP-hard, we cannot find algorithms to approximate MDCP in general graphs. In the next section, we give algorithms for some important special cases.

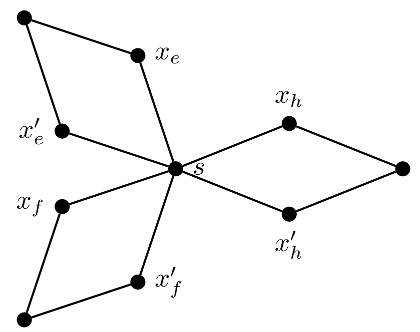
5 Polynomial cases

In this section we give polynomial time algorithms for MDCP and DCP for some important special cases. We recall that, as is MDCP an optimization version of DCP, an algorithm for a problem implies an algorithm for the other one.

In particular, we solve DCP for the cases where the input graph is a Directed Acyclic Graph (DAG), for the cases where the number of colors is bounded by a constant (i.e. $|\mathcal{R}| = O(1)$) and for some cases where the degree of the graph is strictly smaller than 5.



(a) s in G



(b) s in H

Edge	Colors
$\{s, x_e\}$	C_1, C_2
$\{s, x'_e\}$	C_3, C_4
$\{s, x_f\}$	C_1, C_3
$\{s, x'_f\}$	C_2, C_4
$\{s, x_h\}$	C_1, C_4
$\{s, x'_h\}$	C_2, C_3

Figure 8: Color assignment for s.

5.1 Directed acyclic graphs

In this section we give an algorithm for finding $k = O(1)$ color-disjoint paths in a DAG. Note that, the transformations in Theorems 1–3 can be modified in a straightforward way in order to obtain a DAG. It follows that, even in DAGs, finding k color-disjoint paths is NP-hard if k is not a fixed constant and it is not in FPT. Therefore, we will give an algorithm with complexity $O(|V|^{O(poly(k))})$, note that this is the best possible according to the lower bounds.

First we give an algorithm for finding 2 color-disjoint paths and then we generalize it to the case of $k = O(1)$. The algorithm uses two graph transformations introduced in [13] and [10] to find one shortest path in a multi-colored graph with the star property and to find two vertex-disjoint paths with forbidden pairs in a layered graph, respectively. We first give some definitions.

Definition 7 (Layered graph) *A directed graph $G = (V, E)$ is layered if there is a layering function $l : V \to [0, 1, \ldots, (|V| - 1)]$ such that for every arc $(u, v) \in E$, $l(v) = l(u) + 1$. We say that vertex u is in layer $l(u)$ and arc (x, y) is in layer $l(x)$. Layered directed graphs are acyclic.*

Definition 8 (Vertex-disjoint paths with forbidden pairs) *Given a directed graph $G = (V, E)$, two vertices s and t and a set of pairs of arcs $F \subseteq E \times E$ (called forbidden pairs), find two paths P_1 and P_2 between s and t such that P_1 and P_2 are vertex-disjoint and for any forbidden pair $\{e_1, e_2\} \in F$, if P_1 uses e_1 (e_2, resp.), then P_2 does not use e_2 (e_1, resp.).*

A topological ordering of a DAG $G = (V, E)$ is a linear ordering of its vertices such that, for every arc $(u, v) \in E$, u comes before v in the ordering. We define the ordering function $f_o : V \to \mathbb{N}$ as follows:

$$f_o(v) = \begin{cases} 0 & \text{if } \Gamma^-(v) = \emptyset \\ \max\{f_o(u) \mid u \in \Gamma^-(v)\} + 1 & \text{otherwise.} \end{cases}$$

The algorithm is based on the following observations.

- Every DAG can be transformed in polynomial time to a layered DAG. In fact, we can order topologically the DAG, and then replace every arc (u, v) with a directed path from u to v of length $f_o(v) - f_o(u)$, with f_o being the ordering function.

- The graph transformation introduced in [13] (see pages 6–7) to find a shortest colored-path in graphs with the star property can be used to find two color-disjoint paths in a multi-colored graph.

  Let mG be a multi-colored graph and G_m the graph obtained by applying such a transformation, we will show that finding two color-disjoint paths in G is equivalent to finding two vertex-disjoint paths with forbidden pairs in G_m, with the forbidden pairs being pairs of arcs belonging to the same color. However, the transformation has to be slightly modified to work for directed graphs. We report the modified transformation later in this section and we refer to it as Transformation 1 in the remainder of the section. It is important to note that if the directed graph is layered, the auxiliary graph will be layered as well.

- It has been proved in [10] that finding two vertex-disjoint paths with forbidden pairs can be solved in polynomial time in layered DAGs where for every forbidden pair $\{e_1, e_2\}$, the arcs e_1 and e_2 are in the same layer of the graph. In [10] (pages 119–120), the two paths have distinct sources and sinks, but the algorithm can be adapted to find two paths from s to t. If $G = (V, E)$ is a layered graph, l its layering function and F is a set of forbidden pairs and we want to find two vertex-disjoint paths with forbidden pairs from s to t, the transformed graph $G' = (V', E')$ will be constructed as follows:

$$V' = \{< u, v > |(u, v) \in V^2, u \neq v, \text{ and } l(s) < l(u) = l(v) < l(t)\} \cup \{s\} \cup \{t\}$$

$$\begin{aligned} E' = \{(< u_1, v_1 >, < u_2, v_2 >)|((u_1, u_2), (v_1, v_2)) \in E \times E - F\} \\ \cup\{(s, < u, v >)|((s, u), (s, v)) \in E \times E - F\} \\ \cup\{(< u, v >, t)|((u, t), (v, t)) \in E \times E - F\} \end{aligned}$$

  In [10] it has been proven that finding 2 vertex-disjoint pair in G with forbidden pairs F corresponds to finding a path in G'. We will refer to this transformation as transformation 2 in the remainder of this paper.

The algorithm for finding 2 color-disjoint paths is the following.

1. Topologically sort G and define the ordering function f_o;

2. For every arc a from v_1 to v_2 such that $f_o(v_2) - f_o(v_1) > 1$:

   - replace a with a directed path $P = (r_1, \ldots, r_{f_o(v_2)-f_o(v_1)})$ from v_1 to v_2;
   - assign to r_1 the colors of a incident to v_1 and to $r_{f_o(v_2)-f_o(v_1)}$ the colors of a incident to v_2;

   We obtain then a layered DAG G_1 with the star property.

3. Apply transformation 1 to G_1.

1. To each color we associate the center of the star of the edges with this color. If the color has only one occurrence we choose arbitrarily as associated center one of the end vertices of the edge containing this color.
2. To a multicolored digraph mG we associate an auxiliary digraph G_{st} (which is almost the line digraph of mG) as follows.
   - 2.1. To each arc e of mG we associate in G_{st} a vertex e and we join two vertices e and e' by an arc from e to e', if the terminal vertex x of e is the initial vertex of e'.
   - 2.2. In G_{st} we give to the arc (e, e') the colors of e and e' having x as center of their star.
   - 2.3. We add two vertices s and t and join s (resp. t) to the vertices e (resp. f) associated to an arc e (resp. f) having s (resp. t) as initial vertex (resp. terminal vertex) and give to the arc (s, e) (resp. (e, t)) the colors of e (resp f) having s (resp. t) as center.

Figure 9: Transformation 1

Let $G_2 = (V_2, E_2)$ be the obtained graph. Let F be the set of forbidden pairs of G_2, which are the pairs of arcs of the same color. Note that G_2 is a layered graph as well and the edges of a forbidden pairs are in the same layer.

4. Apply the graph transformation 2 to G_2. The new graph is G_3.

$$V_3 = \{< u, v > |(u, v) \in V_2^2, u \neq v, \text{ and } l(s) < l(u) = l(v) < l(t)\} \cup \{s\} \cup \{t\}$$

$$\begin{aligned} E_3 = \{(< u_1, v_1 >, < u_2, v_2 >)|((u_1, u_2), (v_1, v_2)) \in E_2 \times E_2 - F\} \\ \cup\{(s, < u, v >)|((s, u), (s, v)) \in E_2 \times E_2 - F\} \\ \cup\{(< u, v >, t)|((u, t), (v, t)) \in E_2 \times E_2 - F\} \end{aligned}$$

Figure 10: Transformation 2

5. A path from s to t in G_3 corresponds to two vertex-disjoint paths with forbidden pairs F in G_2 which correspond to two color-disjoint paths in G_1 and then G.

The algorithm can be generalized to find k color-disjoint paths using the same graph transformation with the following modifications:

- F is the set of forbidden k-tuples (a k-tuple of arcs is forbidden if two of its elements have a color in common).
- G_3 is constructed as follows:

– In each layer of G_2, every k-tuple of vertices is represented by a vertex in G_3

– Two vertices in G_3 are linked by an arc if there is a k-tuple r of arcs linking the corresponding two k-tuples of vertices, such that r is not in F.

The next theorem shows the correctness of the above algorithm.

Theorem 5 *There exists an algorithm that solves* `DCP` *if in a DAG which requires* $O((|V|\Delta)^{2k})$ *computational time.*

Proof. We show that each transformation is correct.

It is straightforward to see that there exist k color-disjoint paths in G if and only if there exist k color-disjoint paths in G_1.

We now show that there exist k color-disjoint paths in G_1 if and only if there exist k vertex-disjoint paths in G_2 with forbidden k-tuples F. Graph G_1 has k color-disjoint paths $P_1, P_2, \ldots, P_k$ from s to t, where $P_i = \{s, v_{i,1}, v_{i,2}, \ldots, v_{i,j_i}, t\}$, for each $i = 1, 2, \ldots, k$ if and only if the paths $P'_i = \{s, (s, v_1), (v_1, v_2), \ldots, (v_{i,j_i}, t), t\}$ in G_2 are vertex-disjoint (as paths P_i are edge-disjoint) and they do not use a forbidden k-tuple (as paths P_i are color-disjoint).

We now show that there exist k vertex-disjoint paths in G_2 with forbidden k-tuples F if and only if there exists one path from s to t in G_3. Since the input graph is a layered graph and the edges of a forbidden pairs are in the same layer this follows by the same arguments used in [10] (see page 119) but with k-tuples instead of pairs and only one source-destination pair.

Step 1 of the algorithm requires $O(|V| + |E|)$ time. Step 2 requires $O(|V|^2)$ as, in the worst case, G_1 has $O(|V|^2)$ vertices and $O(|V|^2)$ arcs. Step 3 adds $O(\Delta^2)$ arcs for each vertex and a vertex for each arc of G_1 and hence G_2 has $O(|V|^2)$ vertices and $O((|V|\Delta)^2)$ arcs. Therefore, Step 3 requires $O((|V|\Delta)^2)$ time. In G_3 there are $O(|V|^{2k})$ vertices and $O((|V|\Delta)^{2k})$ edges. Hence Step 3 requires $O((|V|\Delta)^{2k})$ time. □

In the following, we give an example of how the algorithm finds 2 color-disjoint paths. Let's consider the DAG in Figure 11 and the ordering function f_o defined as: $f_o(s) = 0$, $f_o(a) = f_o(b) = f_o(c) = 1$, $f_o(d) = 2$, $f_o(e) = f_o(f) = 3$ and $f_o(t) = 4$. Figures 12, 13 and 14 represent the graphs G_1, G_2 and, G_3 obtained after applying Steps 2, 3, and 4 of the algorithm, respectively.

The only path from s to t in G_3 is $P = (s, v_2v_3, v_5v_6, v_8v_9, v_{11}v_{12}, t)$. In G_2, P corresponds to the two vertex-disjoint paths with forbidden pairs $P_1 = (s, v_2, v_5, v_9, v_{12}, t)$ and $P_2 = (s, v_3, v_6, v_8, v_{11}, t)$, which correspond in G to the two color-disjoint paths (s, b, d, f, t) and (s, c, d, e, t).

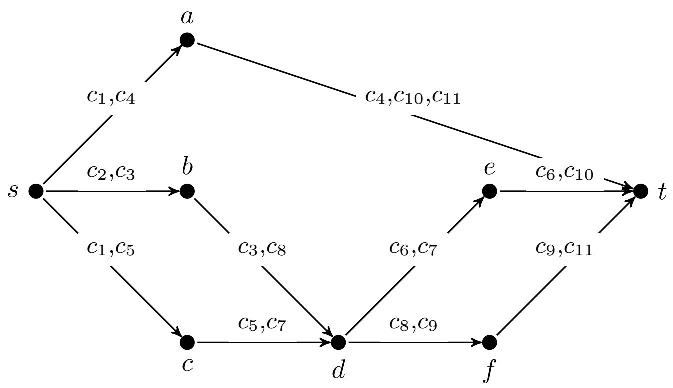


Figure 11: $G = (V, E)$

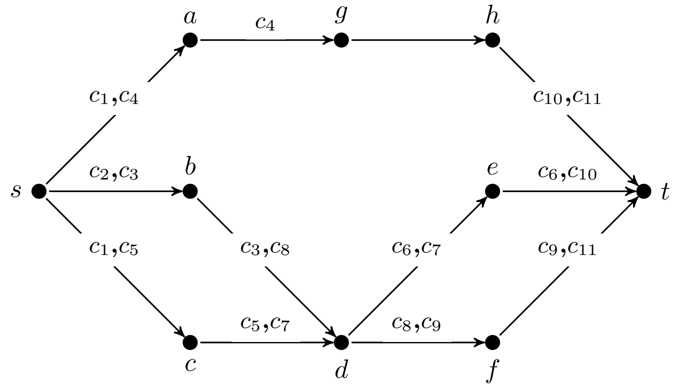


Figure 12: $G_1 = (V_1, E_1)$ after applying the Step 2 of the algorithm (Layering).

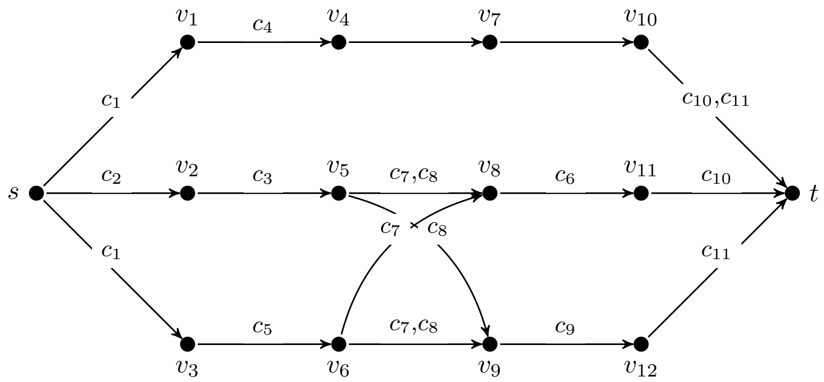


Figure 13: $G_2 = (V_2, E_2)$ after applying the Step 3 of the algorithm (Transformation 1). The forbidden pairs are: $F = \{\{(s, v_1), (s, v_3)\}; \{(v_5, v_8), (v_5, v_9)\}; \{(v_5, v_8), (v_6, v_8)\}; \{(v_5, v_8), (v_6, v_9)\}; \{(v_5, v_9), (v_6, v_9)\}; \{(v_6, v_8), (v_6, v_9)\}; \{(v_{10}, t), (v_{11}, t)\}; \{(v_{10}, t), (v_{12}, t)\}\}$.

5.1.1 Optimal diverse routing in weighted DAGs with the star property.

The above algorithm can be also used to find an minimum cost pair of color-disjoint paths in a weighted multi colored DAG. Let us assume that w is a weight function on the arcs of a graph G.

The previous algorithm can be adapted to find a minimum cost pair of color-disjoint paths by applying the following modifications.

1. In step 2, assign to (r_1, r_2) the weight of (v_1, v_2);

2. In step 1 of transformation 1, assign to (x, v_m) the weight of (x, y);

3. In step 2 of transformation 1, assign to (w, w') the sum of the weights of (w, v) and (v, w');

4. In transformation 2, assign to every arc of E_3 the sum of the weights of the two corresponding arcs in E_2;

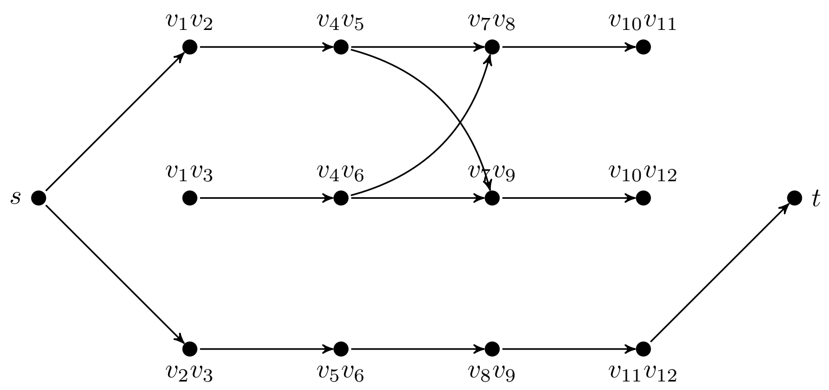


Figure 14: $G_3 = (V_3, E_3)$ after applying the Step 4 of the algorithm (Transformation 2).

5. With these modifications, the shortest path in G_3 corresponds to the optimal pair of disjoint paths in G.

5.1.2 Finding 2 color-disjoint paths with the minimum total number of colors in DAGs with the star property

We can use the previous algorithm to find a pair of color-disjoint paths with the minimum total number of colors by applying the following modifications:

1. In step 4, assign to every new edge $(< u_1, v_1 >, < u_2, v_2 >)$ a weight equal to $|\mathcal{R}((u_1, v_1)) \cup \mathcal{R}((u_2, v_2))|$

2. The shortest path in the weighted graph G_3 will correspond then to the pair of color-disjoint paths with the minimum number of colors.

5.2 Bounded number of colors

In this section, we give an algorithm for finding k color-disjoint paths in the special case where the number $|\mathcal{R}|$ of colors in the network is bounded by a constant, i.e. $|\mathcal{R}| = O(1)$. We observe that such an algorithm works for every graph topology and even if the star property does not hold.

The algorithm is as follows.

1. Enumerate all the subsets of $\mathcal{R}$. Let F be the family of all the subsets of $\mathcal{R}$.

2. For each set f in F consider the subgraph G_f of G induced by all the arcs e such that $\mathcal{R}(e) \subseteq f$.

3. Remove from F all the elements f such that G_f does not have a path from s to t and associate to the other elements a path P_f from s to t.

4. Build a k-Set Packing instance where $X = \mathcal{R}$ and $C = F$ and solve such instance by the exact algorithm in [4]. If there exists a tuple of k color-disjoint sets in F, then output the set of paths corresponding to such tuple; otherwise mG does not contain k color-disjoint paths.

The correctness and complexity of the above algorithm are shown in the next theorem.

Theorem 6 *There exists an algorithm that solves* DCP *if* $|\mathcal{R}| = O(1)$ *which requires* $O(|V|+|E|)$ *computational time.*

Proof. We prove the correctness by contradiction. First let us assume that the algorithm outputs k paths $P_1, P_2, \ldots, P_k$ but they are not color-disjoint. These implies that there exists two paths P_i and P_j that share some colors, that is the corresponding set in C in the set packing instance of Step 4 are not disjoint, a contradiction.

Let us assume that the algorithm does not output the paths but that there exist k disjoint paths in G. Let us consider k disjoint paths $P_1, P_2, \ldots, P_k$ in G and let $C_1, C_2, \ldots, C_k$ be the sets of colors induced by them. Then these sets belongs to F at step 1 and are not removed from it at step 3 because for each C_i there exists at least a path (e.g. P_i) from s to t. Moreover, as sets $C_1, C_2, \ldots, C_k$ are disjoint, will be chosen at step 4 by the set packing algorithm.

The family F contains at most $2^{|\mathcal{R}|} = O(1)$ elements and hence the step 1, 2 and 3 require $O(1)$, $O(|V|+|E|)$ and $O(|V|+|E|)$ computational time, respectively. Moreover, the algorithm in [4] requires $O(|C|2^{|X|}|X|^{O(1)})$ time and then, as $|X| = |\mathcal{R}| = O(1)$ and $|C| = |F| \leq 2^{|\mathcal{R}|} = O(1)$, it requires $O(1)$ time. It follows that the algorithm requires $O(|V|+|E|)$ overall time. □

5.3 Bounded degree

In this section, we give algorithms for finding k color-disjoint paths when $\Delta < 4$ and for finding 2 color-disjoint paths when $\Delta = 4$. First, note that the maximum number of color-disjoint paths in a graph is upper bounded by Δ.

If $\Delta \leq 2$ the problem is trivial as the graph is either a path or a cycle. In the first case, it always exists only one path from s and t. In the second case, the only vertices where the two possible paths can share colors are s and t and hence it is enough to check if the two arcs incident to s (and t) are color-disjoint.

If $\Delta = 3$, observe that if two different paths share a vertex, they necessarily share also an arc and hence all the colors of that arc. It follows that DCP can be solved by finding $k \in \{2,3\}$ vertex-disjoint paths from all the subsets of size k of the neighbors of s to all the subsets of size k of the neighbors of t. In detail, let S be the subsets of size k of $\Gamma^+(s)$ such that for each $x \in S$ and for each $s_i, s_j \in x$, $\mathcal{R}((s,s_i)) \cap \mathcal{R}((s,s_j)) = \emptyset$ and let T be the subsets of size k of $\Gamma^-(t)$ such that for each $y \in T$ and for each $t_i, t_j \in y$, $\mathcal{R}((t_i,t)) \cap \mathcal{R}((t_j,t)) = \emptyset$. Then, there exist k color-disjoint paths from s to t if and only if there exists k vertex-disjoint paths from a subset x of S to a subset y of T. As finding $k \in \{2,3\}$ vertex-disjoint paths requires linear time [23, 24], then the above algorithm requires $O(\Delta^{2k} \cdot (|V|+|E|)) = O(V|+|E|)$ time.

If $\Delta = 4$, the algorithm for solving 2-DCP is reported in Figure 15.

The correctness of the above algorithm is shown in the next theorem.

Theorem 7 *There exists an algorithm that solves* DCP *if* $\Delta = 4$ *which requires* $O(|V|+|E|)$ *computational time.*

Proof. If there exist two neighbors s_1 and s_2 of s and two neighbors t_1 and t_2 of t such that $\mathcal{R}((s,s_1)) \cap \mathcal{R}((s,s_2)) = \emptyset$ and $\mathcal{R}((t_1,t)) \cap \mathcal{R}((s_2,t)) = \emptyset$, and there exists two vertex-disjoint paths from $\{s_1,s_2\}$ to $\{t_1,t_2\}$ in G', then such two paths are clearly color-disjoint in G.

Otherwise, for each pair $\{s_1,s_2\}$ of neighbors of s and for each pair $\{t_1,t_2\}$ of neighbors of t such that $\mathcal{R}(\{s,s_1\}) \cap \mathcal{R}(\{s,s_2\}) = \emptyset$ and $\mathcal{R}((t_1,t)) \cap \mathcal{R}((s_2,t)) = \emptyset$ there exists at least one cut vertex in G'. If one of such cut vertices has degree 2 or 3, then each path from $\{s_1,s_2\}$ to $\{t_1,t_2\}$ has to share an edge and hence cannot exist color-disjoint paths from $\{s_1,s_2\}$ to $\{t_1,t_2\}$. It follows that there exists two color-disjoint paths in G only if all the cut-vertices from some pair

Consider the graph G' obtained from G by removing vertices s and t and the arcs incident to them;

foreach *pair* $\{s_1, s_2\}$ *of neighbors of* s *in* G *such that* $\mathcal{R}((s, s_1)) \cap \mathcal{R}((s, s_2)) = \emptyset$ *and for each pair* $\{t_1, t_2\}$ *of neighbors of* t *such that* $\mathcal{R}((t_1, t)) \cap \mathcal{R}((t_2, t)) = \emptyset$*:* **do**

  if *there exist 2 vertex-disjoint paths form* $\{s_1, s_2\}$ *to* $\{t_1, t_2\}$ *in* G' **then**

    There exist two color-disjoint paths from s to t in G;

  else

    if *all the cut-vertices from* $\{s_1, s_2\}$ *to* $\{t_1, t_2\}$ *in* G' *have degree 4* **then**

      if *for each cut vertex* v *there exist two arcs* e *and* f *ingoing to* v *which are closer to* s *and the other two arcs* g *and* h *outgoing from* v *which are closer to* t *such that* $\mathcal{R}(e) \cap \mathcal{R}(f) = \emptyset$*,* $\mathcal{R}(g) \cap \mathcal{R}(h) = \emptyset$ $\mathcal{R}(e) \cap \mathcal{R}(h) = \emptyset$*, and* $\mathcal{R}(f) \cap \mathcal{R}(g) = \emptyset$ **then**

        There exist two color-disjoint paths from s to t in G;

No 2 color-disjoint paths from s to t exist in G;

Figure 15: Algorithm for solving k-DCP when $\Delta = 4$.

$\{s_1, s_2\}$ to some pair $\{t_1, t_2\}$ in G' have degree 4. Moreover, if there exist two color-disjoint paths from some pair $\{s_1, s_2\}$ to some pair in G' $\{t_1, t_2\}$ then for all the cut-vertices v from $\{s_1, s_2\}$ to $\{t_1, t_2\}$, there exist two arcs e and f ingoing to v which are closer to s and the other two arcs g and h outgoing from v which are closer to t such that $\mathcal{R}(e) \cap \mathcal{R}(f) = \emptyset$, $\mathcal{R}(g) \cap \mathcal{R}(h) = \emptyset$ $\mathcal{R}(e) \cap \mathcal{R}(h) = \emptyset$, and $\mathcal{R}(f) \cap \mathcal{R}(g) = \emptyset$. The complexity is given by the complexity of finding two vertex-disjoint st-paths from each pair of neighbors of s to each pair of neighbors of t, that is $O(\Delta^4(|V| + |E|)) = O(|V| + |E|)$. □

Note that the above algorithm cannot be extended neither to find 3 or 4 color-disjoint paths on a graphs with $\Delta = 4$ nor to the case of $\Delta \geq 5$.

References

[1] Giorgio Ausiello, Alessandro D'Atri, and Marco Protasi. Structure preserving reductions among convex optimization problems. *J. Comput. Syst. Sci.*, pages 136–153, 1980.

[2] P. Berman and T. Fujito. On approximation properties of the independent set problem for degree 3 graphs. In S. Akl, F. Dehne, J-R. Sack, and N. Santoro, editors, *4th International Workshop on Algorithms and Data Structures (WADS95)*, volume 955 of *Lecture Notes in Computer Science*, pages 449–460. Springer Berlin / Heidelberg, 1995.

[3] R. Bhandari. *Survivable Networks: Algorithms for Diverse Routing.* Kluwer Academic Publishers, Norwell, MA, USA, 1998.

[4] A. Björklund, T. Husfeldt, and M. Koivisto. Set partitioning via inclusion-exclusion. *SIAM J. Comput.*, 39(2):546–563, 2009.

[5] H. Broersma, X. Li, G. Woeginger, and S. Zhang. Paths and cycles in colored graphs. *Australas. J. Combin.*, 31:299–311, 2005.

[6] F. Carrabs, R. Cerulli, and M. Gentili. The labeled maximum matching problem. *Computers & Operations Research*, 36(6):1859–1871, 2009.

[7] R. Cerulli, P. Dell'Olmo, M. Gentili, and A. Raiconi. Heuristic approaches for the minimum labelling hamiltonian cycle problem. *Electronics Notes in Discrete Mathematics*, 25:131–138, 2006.

[8] R. Cerulli, A. Fink, M. Gentili, and S. Voß. *The Next Wave on Computing, Optimization, and Decision Technologies*, chapter Metaheuristics comparison for the minimum labelling spanning tree problem, pages 93–106. Springer, New York, b.l. golden and s. raghavan and e.a. wasil edition, 2005.

[9] R. Cerulli, A. Fink, M. Gentili, and S. Voß. Extensions of the minimum labelling spanning tree problem. *Journal of Telecommunications and Information Technology*, 4:39–45, 2006.

[10] V. T. Chakaravarthy. New results on the computability and complexity of points–to analysis. In *Proceedings of the 30th ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, POPL '03, pages 115–125. ACM, 2003.

[11] R. Chang and S. Leu. The minimum labeling spanning trees. *Information Processing Letters*, 63(5):277–282, 1997.

[12] D. Coudert, P. Datta, S. Perennes, H. Rivano, and M-E. Voge. Shared risk resource group: Complexity and approximability issues. *Parallel Processing Letters*, 17(2):169–184, June 2007.

[13] D. Coudert, S. Perennes, H. Rivano, and M-E. Voge. Shared risk resource groups and colored graph: Polynomial and FPT optimisation problems. Submitted.

[14] B. Couëtoux, L. Gourvès, J. Monnot, and O. A. Telelis. Labeled traveling salesman problems: Complexity and approximation. *Discrete Optimization*, 7(1-2):74–85, February 2010.

[15] P. Datta and A.K. Somani. Graph transformation approaches for diverse routing in shared risk resource group (SRRG) failures. *Computer Networks*, 52(12):2381–2394, August 2008.

[16] J. Doucette and W. Grover. Shared-risk logical span groups in span-restorable optical networks: Analysis and capacity planning model. *Photonic Network Communications*, 9(1):35–53, 2005.

[17] Rod G. Downey and Michael R. Fellows. Fixed-parameter tractability and completeness ii: On completeness for w[1]. *Theoretical Computer Science*, 141(1-2):109 – 131, 1995.

[18] A. Faragó. A graph theoretic model for complex network failure scenarios. In *8th INFORMS Telecommunications Conference*, Dallas, Texas, March 2006.

[19] M. R. Fellows, J. Guob, and I. Kanj. The parameterized complexity of some minimum label problems. *Journal of Computer and System Sciences*, 76(8):727–740, December 2010.

[20] M. Garey and D. Johnson. *Computers and Intractability: A Guide to the theory of NP-completeness*. Freeman NY, 1979.

[21] R. Hassin, J. Monnot, and D. Segev. Approximation algorithms and hardness results for labeled connectivity problems. *Journal of Combinatorial Optimization*, 14(4):437–453, November 2007.

[22] R. Hassin, J. Monnot, and D. Segev. The complexity of bottleneck labeled graph problems. In *33rd International Workshop on Graph-Theoretic Concepts in Computer Science (WG)*, volume 4769 of *LNCS*, pages 328–340, Dornburg near Jena, Germany, June 2007. Springer.

[23] J. E. Hopcroft and R. E. Tarjan. Algorithm 447: efficient algorithms for graph manipulation. *Commun. ACM*, 16(6):372–378, June 1973.

[24] J. E. Hopcroft and R. E. Tarjan. Dividing a graph into triconnected components. *SIAM J. Comput.*, 2(3):135–158, 1973.

[25] J.Q. Hu. Diverse routing in mesh optical networks. *IEEE Transactions on Communications*, 51:489–494, March 2003.

[26] Q. Jiang, D.S. Reeves, and P. Ning. Improving robustness of PGP keyrings by conflict detection. In *RSA Conference Cryptographers' Track (CT-RSA)*, volume 2964 of *LNCS*, pages 194–207. Springer, February 2004.

[27] V. Kann. Maximum bounded 3-dimensional matching is max snp-complete. *Inf. Process. Lett.*, 37(1):27–35, January 1991.

[28] S.O. Krumke and H.-C. Wirth. On the minimum label spanning tree problem. *Information Processing Letters*, 66(2):81–85, 1998.

[29] X. Luo and B. Wang. Diverse routing in WDM optical networks with shared risk link group (SRLG) failures. In *Proc. DRCN*, pages 445–452. IEEE, October 2005.

[30] R. Niedermeier. *Invitation to fixed-parameter algorithms.* Oxford lecture series in mathematics and its applications. Oxford University Press, 2006.

[31] L. Shen, X. Yang, and B. Ramamurthy. Shared risk link group (SRLG)-diverse path provisioning under hybrid service level agreements in wavelength-routed optical mesh networks. *IEEE/ACM Transactions on Networking*, 13:918– 931, August 2005.

[32] S. Szeider. Finding paths in graphs avoiding forbidden transitions. *Discrete Applied Mathematics*, 126(2–3):261 – 273, 2003.

[33] Y. Wan, G. Chen, and Y. Xu. A note on the minimum label spanning tree. *Information Processing Letters*, 84(2):99–101, 2002.

[34] S. Yuan, S. Varma, and J.P. Jue. Minimum-color path problems for reliability in mesh networks. In *Proc. IEEE INFOCOM*, volume 4, pages 2658–2669, Houston, TX, USA, March 2005.

RESEARCH CENTRE
SOPHIA ANTIPOLIS – MÉDITERRANÉE

2004 route des Lucioles - BP 93
06902 Sophia Antipolis Cedex

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr

ISSN 0249-6399